基于远距离标签评论和细粒度的推荐算法

Jianmo Ni, Jiacheng Li, Julian McAuley

摘要：近期的一些研究考虑了生成评论（或“提示”）的问题，作为解释为什么一个推荐可能符合用户兴趣的一种形式。尽管这一途径很有前景，但我们展示了现有方法在生成与用户决策过程相关的理由（无论是质量还是内容方面）时都存在困难。我们旨在引入新的数据集和方法来解决这个推荐理由任务。在数据方面，我们首先提出了一种“抽取式”方法，用于识别能够证明用户意图的评论段落；然后使用这种方法远程标记大量的评论语料库，并构建大规模的个性化推荐理由数据集。在生成方面，我们利用这些数据设计了两种个性化生成模型：（1）一种带有方面规划的基于引用的 Seq2Seq 模型，能够生成涵盖不同方面的理由，以及（2）一种基于方面条件的掩码语言模型，能够基于从理由历史中提取的模板生成多样化的理由。我们在两个真实世界数据集上进行了实验，结果显示我们的模型能够生成令人信服且多样化的理由。

关键词：推荐算法，语料库，数据集，Seq2Seq 模型，掩码语言模型

1 引言

向用户解释或证明推荐的合理性有可能提高其透明度和可靠性。然而，提供有意义的解释仍然是一项困难的任务，部分原因是许多推荐模型的“黑盒”性质，但也因为我们根本缺乏指定什么是“好”的理由的基准数据集。

先前的工作试图从大量评论中学习用户偏好和写作风格[10][24]（Dong et al., 2017; Ni and McAuley, 2018）。

表 1.1 与评论和提示相比，我们寻求自动生成更简洁、具体且对决策有帮助的推荐理由。

评论示例：
我爱这个小摊位！**椰子摩卡冷饮和焦糖玛奇朵都很美味。** 哇，真是个特别的发现。我和我丈夫有过的最独特而特别的约会之夜之一。
提示示例：
食物很好。氛围很好。诺奇面非常好吃。 我对这个地方简直欲罢不能。
理由示例：
食物份量很大。 **普通的奶酪玉米饼非常好吃且价格很便宜。**

来自评论、提示和我们标注数据集的理由示例以粗体标出，以自然语言形式生成解释，例如生成类似于用户会对产品写的综合评论。然而，大量的评论文本（或“提示”文本）对大多数用户的决策制定常常关联不大（例如，它们描述了冗长的经历或一般性

的背书），并且可能不适合作为解释使用，无论是内容还是语言风格。因此，直接从评论（或提示）中学习的现有模型可能无法捕获解释用户购买行为的关键信息。表 1.1 显示了评论、提示和理想理由的示例。最近，有研究致力于生成提示的任务，其中提示是评论的简洁总结[19]（Li et al., 2017）。尽管提示简洁，其中一些可能适合作为推荐理由的候选，但只有少数电子商务系统提供伴随评论的提示。即使在提供提示的系统中，提示的数量通常也远小于评论的数量。因此，这些方法在用户互动高度稀疏的设置中遭受普遍性问题。

另一方面，在如理由生成等个性化内容生成场景中生成多样化的响应是必不可少的。与其总是预测最流行的理由，不如为基于他们个人兴趣的不同用户呈现多样化的理由。最近的工作表明，将先验知识纳入生成框架可以极大地提高多样性。先验知识可以包括故事生成中的故事线[33]（Yao et al., 2019），或对话系统中的历史回应[29]（Weston et al., 2018）。

在这项工作中，我们的目标是生成令人信服且多样化的理由。为了解决缺乏关于“好”的理由的基准数据的挑战，我们提出了一个能够从大量评论或提示中识别理由的流程。我们从理由中提取细粒度的方面，并构建包含一系列代表性方面的用户画像和物品概况。为了提高生成质量和多样性，我们提出了两种生成模型：（1）一种带有方面规划的基于参考的 Seq2Seq 模型，它以之前的理由为参考，可以基于不同方面产生理由；以及（2）一种可以从以前的理由中提取的模板生成多样化理由的方面条件掩码语言模型。

我们的贡献有三方面：

1. 为了促进推荐理由生成，我们提出了一个流程来识别理由候选并从大量评论中构建基于方面的用户画像和物品概况。通过这种方法，我们能够构建大规模的个性化理由数据集。我们在可解释推荐任务中使用这些抽取式理由片段，并展示这些比整个评论更好的训练资源。
2. 我们提出了两种基于参考注意力、方面规划技术和个性条件掩码语言模型的模型。我们展示了添加此类个性化信息使得模型能够生成高质量和多样性的理由。
3. 我们在 Yelp 和 Amazon Clothing 的两个真实世界数据集上进行了广泛的实验。我们提供了一个关于 Yelp 数据集上“好”的理由的标注数据集，并展示了在这个数据集上训练的二分类器能够很好地推广到 Amazon Clothing 数据集。我们研究了不同的解码策略，并比较了它们对生成性能的影响。

2 数据集生成

在本节中，我们介绍了从原始用户评论中提取高质量理由的流程。具体来说，我们的目标是识别可以作为评论使用的评论片段，并基于它们构建一个个性化的理由数据集。我们的流程包括三个步骤：

1. 用二元标签注释一组评论片段，即确定它们是“好”的还是“坏”的理由。

2. 在注释的子集上训练一个分类器，并将其应用于远程标记所有评论片段，以提取每个用户和物品对的“好”理由。
3. 对提取出的理由进行细粒度方面提取，并构建用户画像和物品概况。

2.1 从评论中识别理由

第一步是从评论中提取适合用作理由的文本片段。我们定义每个片段为一个基本话语单元（Elementary Discourse Unit, EDU；[21]Mann and Thompson, 1988），它对应于一系列从句。我们使用 Wang et al. (2018)[28]的模型从评论中获取 EDU。最近的研究表明，EDU 可以提高文档级别摘要[4]（Bhatia et al., 2015）和观点摘要[2]（Angelidis and Lapata, 2018）的性能。

表 2.1 将评论片段分类为推荐理由的好或坏的性能。

方法	F1 值	召回率	精确度
BOW-Xgboost	0.559	0.679	0.475
CNN	0.644	0.596	0.700
LSTM-MaxPool	0.675	0.703	0.650
BERT	**0.747**	0.700	**0.800**
BERT-SA (一轮)	0.481	0.975	0.320
BERT-SA (三轮)	0.491	**1.000**	0.325

在将评论预处理为 EDU 之后，我们分析了推荐理由和评论之间的语言差异，并建立了两条规则来过滤不太可能适合作为理由的片段：（1）含有第一人称或第三人称代词的片段，（2）过长或过短的片段。接下来，两位专家评注员接触到了 1000 个未被过滤掉的片段，并被要求确定它们是否是“好”的理由。标注是迭代进行的，随后进行反馈和讨论，直到两位评注员之间的质量达到一致。在这个过程的最后，二元标签任务（好对坏）的评注员间一致性，通过 Cohen’s kappa（Cohen, 1960）测量[8]，经过对齐后为 0.927。然后，评注员进一步标注了 600 个片段。总体上，24.8%的片段被标记为好。

2.2 自动分类

我们的下一步是将标签传播到完整的评论语料库中。在这里，我们采用 BERT（Devlin et al., 2019）[9]对我们的分类任务进行微调，其中在每个片段的开头添加了一个[CLS]标记，并且将对应于这个标记的最终隐藏状态（即，BERT 的输出）输入到一个线性层以获得二元预测。交叉熵被用作训练损失。

我们将注释的数据集分为训练集、开发集和测试集，比例为 0.8/0.1/0.1，对训练集上的 BERT 分类器进行微调，并在开发集上选择最佳模型。经过三轮微调后，BERT 在测试集上可以达到 0.80 的 F1 分数。我们将 BERT 的性能与多个基线模型进行了比较：（1）一个使用词袋作为句子特征的 XGBoost 模型（2）一个具有三个卷积层和一个线性层的卷积神经网络（CNN）（3）一个具有最大池化层和一个线性层的长短期记忆（LSTM）

网络[13]（Hochreiter and Schmidhuber, 1997）（4）一个在完整的 Yelp 数据集上训练了一轮和三轮的 BERT 情感分类器（BERT-SA）。为了获得 CNN 和 LSTM 模型的预训练词嵌入，我们在 Yelp 评论数据集上应用了 fastText[5]（Bojanowski et al., 2016）。我们将嵌入维度设置为 200，并对其他超参数使用默认值。

表 2.2 展示了我们二元分类任务的结果。BERT 分类器的 F1 分数和精确度高于其他分类器。经过三轮的 BERT-SA 模型仅获得 0.491 的 F1 分数，这确认了情感分析和我们好/坏任务之间的差异，即使片段具有积极的情感，它可能不适合作为理由。

表 2.2 我们标注数据集中带有细粒度方面的理由示例。细粒度方面以斜体和下划线标出。

Yelp：
金枪鱼真的很棒
开胃菜和面食在这里都很出色
甜的和咸的可丽饼都有出色的选择
充满了美味的食物、激动人心的音乐和舞蹈
亚马逊衣服：
材料的质量和材质很好
很棒的衬衫，尤其是考虑到价格
缝合和缝纫做得非常好
很适合作为万圣节的服装

2.3 细粒度方面提取

最后，我们提取每个理由所涵盖的细粒度方面。细粒度方面是用户意见中出现的产品属性。我们采用 Zhang 等人[34]（Zhang et al. (2014)）提出的方法构建一个情感词典，其中包括整个数据集中的一组细粒度方面。然后，我们使用简单的规则来确定每个理由中出现的方面[1]。

表 2.2 展示了我们数据集中的一组示例。每个示例由用户关于某个项目所写的理由组成，并且在理由中提到了多个细粒度方面。**注意，**我们只标注了 Yelp 数据集，在该数据集上训练了分类器，并将模型应用于 Yelp 和 Amazon Clothing 数据集。如表 2.2 所示，训练有素的分类器在两个数据集上都工作得很好。

3 方法

3.1 问题定义

对于每个用户u（或物品i），我们构建一个理由引用集合$D = \{d_1, \ldots, d_l\}$，包含用户

[1] 对于每个方面，如果其单数或复数存在于标记化的理由中，那么我们就认为该方面存在于该理由中。

编写的（或关于物品的）理由，其中l_r是理由的最大数量。我们还根据用户（或物品）之前的理由所涵盖的细粒度方面，获取一个用户画像（或物品概况）$A = \{a_1, \dots, a_K\}$，其中K是方面的最大数量。

给定一个用户u和一个物品i，以及他们的理由引用D_u和D_i，以及u的画像A_u和i的概况A_i，我们的目标是预测理由$J_{u,i} = \{w_1, w_2, \dots, w_T\}$，这些理由解释了为什么物品$i$符合用户$u$的兴趣，其中$T$是理由的长度。

3.2 基于引用的 Seq2Seq 模型

我们的基本模型遵循标准的 Seq2Seq[26]（Sutskever et al., 2014）模型的结构。我们的框架，称为“Ref2Seq”，将用户和物品的历史理由视为引用，并从中学习潜在的个性化特征。图 3-1 展示了我们基于引用的 Seq2Seq 模型的结构。它包括两个组件：（1）两个序列编码器，通过取之前的理由作为引用来学习用户和物品的潜在表示；（2）一个序列解码器，结合用户和物品的表示来生成个性化的理由。

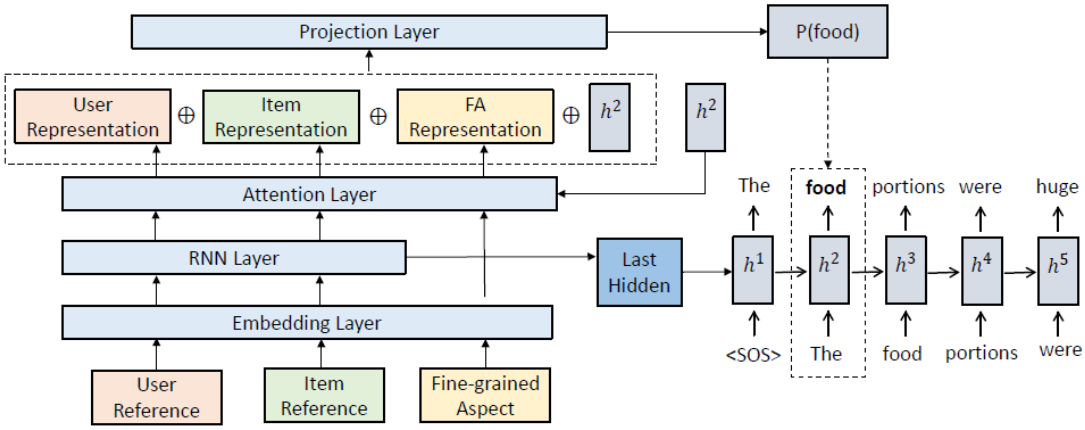

图 3-1 利用远距离标签评论和细粒度方面为推荐提供依据

序列编码器。我们的用户编码器和序列编码器共享相同的结构，其中包括一个嵌入层、一个双层双向 GRU[7]（Cho et al., 2014）和一个投影层。输入是用户（或物品）引用D，包含一系列历史理由。这些理由通过一个词嵌入层，然后通过 GRU，产生一系列隐藏状态 $e \in \mathbb{R}^{l_s \times l_r \times n}$:

$$\mathrm{E} = \mathrm{Embedding}(D), e = \mathrm{GRU}(\mathrm{E}) = \vec{e} + \overleftarrow{e}$$

其中l_s表示序列的长度，n是编码器 GRU 的隐藏大小， $\mathrm{E} \in \mathbb{R}^{l_s \times l_r \times n}$是嵌入的序列表示，$\vec{e}$和$\overleftarrow{e}$分别是前向和后向 GRU 产生的隐藏向量。

为了结合来自不同“引用”的信息（即，理由），隐藏状态随后通过线性层投影：

$$\hat{e} = W_e \cdot e + b_e$$

其中$\hat{\mathrm{e}} \in \mathbb{R}^{l_s \times n}$是编码器的最终输出，$W_e \in \mathbb{R}^{l_r}, b_e \in \mathbb{R}$是学习到的参数。

序列解码器。解码器是一个双层 GRU，给定一个开始标记预测目标词。解码器的隐藏状态使用用户和物品编码器的最后一个隐藏状态之和初始化。在时间步t，隐藏状态通过基于先前的隐藏状态和输入词的 GRU 单元更新。具体来说：

$$h_0 = e_{l_s}^u + e_{l_s}^i, h_t = \text{GRU}(w_t, h_{t-1})$$

其中$e_{l_s}^u$和$e_{l_s}^i$分别是用户和物品编码器输出$\hat{e}_u$和$\hat{e}_i$的最后隐藏状态。

为了探索引用和生成之间的关系，我们应用一个注意力融合层来总结每个编码器的输出。对于用户和物品引用编码器，注意力向量定义为：

$$a_t^1 = \sum_{j=1}^{l_s} \alpha_{tj}^1 e_j$$

$$\alpha_{tj}^1 = \exp(\tanh(v_\alpha^{1\mathrm{T}}(W_\alpha^1[e_j; h_t] + b_\alpha^1)))/Z;$$

其中$\alpha_{tj}^1 \in \mathbb{R}^n$是时间步$t$上序列编码器的一个注意力向量，$\alpha_{tj}^1$是编码器隐藏状态$e_j$和解码器隐藏状态$h_t$之上的注意力得分，$Z$是归一化项。

分区规划生成。生成理由的一个挑战是如何提高可控性，即直接操纵正在生成的内容。受到“计划并写”[33]（Yao et al., 2019）的启发，我们将基本模型扩展到一个方面规划的 Ref2Seq（AP-Ref2Seq）模型，在生成之前规划一个细粒度方面。这种方面规划可以被视为一种额外的监督形式，而不是一个硬约束，使理由生成更加可控。

当为用户u和物品i生成理由时，我们首先提供一个细粒度方面a作为计划。方面a被输入到词嵌入层以获得方面嵌入E_a。然后，我们计算方面嵌入和解码器隐藏状态之间的得分如下：

$$a_t^2 = \alpha_t^2 E_a$$

$$\alpha_t^2 = \exp(\tanh(v_\alpha^{2\mathrm{T}}(W_\alpha^2[E_a; h_t] + b_\alpha^2)))/Z;$$

其中$\alpha_t^2 \in \mathbb{R}^n$是一个注意力向量，$a_t^2$是一个注意力得分。

用户u的注意力向量a_{ut}^1，物品i的注意力向量a_{it}^1，和细粒度方面a的注意力向量a_t^2，与时间步t的解码器隐藏状态拼接，并投影以获得输出词分布P。时间步t的词w的输出概率由下式给出：

$$p(w_t) = \tanh(W_1[h_t; a_{ut}^1; a_{it}^1; a_t^2] + b_1);$$

其中w_t是时间步t的目标词。给定每个时间步t的概率$p(w_t)$，模型使用与真实序列相比的交叉熵损失进行训练。

3.3 面向方面的条件遮蔽语言模型

尽管基于 Seq2Seq 的模型可以实现高质量输出，但它们往往无法生成多样化内容。近期自然语言生成（NLG）的研究尝试将生成方法与信息检索技术结合起来，以增加生

成内容的多样性[16]（Li et al., 2018; Baheti et al., 2018）。基本思想遵循“检索-编辑”范式，即首先检索历史回应作为模板，然后编辑这些模板生成新内容。由于我们的数据带有细粒度方面的注释，它自然适合于这种“检索-编辑”范式。与此同时，遮蔽语言模型在语言建模方面展现出了优异的性能。近期的研究[27][22]（Wang and Cho, 2019; Mansimov et al., 2019）表明，通过从遮蔽语言模型（例如 BERT）中抽样，可以生成连贯的句子。

受这些工作的启发，我们希望将这种方法扩展成一个条件版本——我们探索使用面向方面的条件遮蔽语言模型（ACMLM）来生成多样化的个性化理由。图 2 展示了我们的面向方面的条件遮蔽语言模型的结构。对于用户u关于项目i所写的理由$J_{u,i}$，我们将预训练的 BERT 模型[9]（Devlin et al., 2019）改造成一个编解码器网络，包括（1）一个方面编码器，它将用户个人特征和项目简介编码成潜在的表示，以及（2）一个遮蔽语言模型序列解码器，它接收一个被遮蔽的理由并预测被遮蔽的词语。

方面编码器。我们的方面编码器使用与 BERT 相同的 WordPiece 嵌入[31]（Wu et al., 2016）。编码器将用户个人特征和项目简介中的细粒度方面交集$A_{ui} = \{a_1, \dots, a_{K'}\}$输入嵌入层，并获得方面嵌入$A_{ui} \in \mathbb{R}^{K' \times n}$，其中$K'$是共同细粒度方面的数量，$n$是 WordPiece 嵌入的维度。

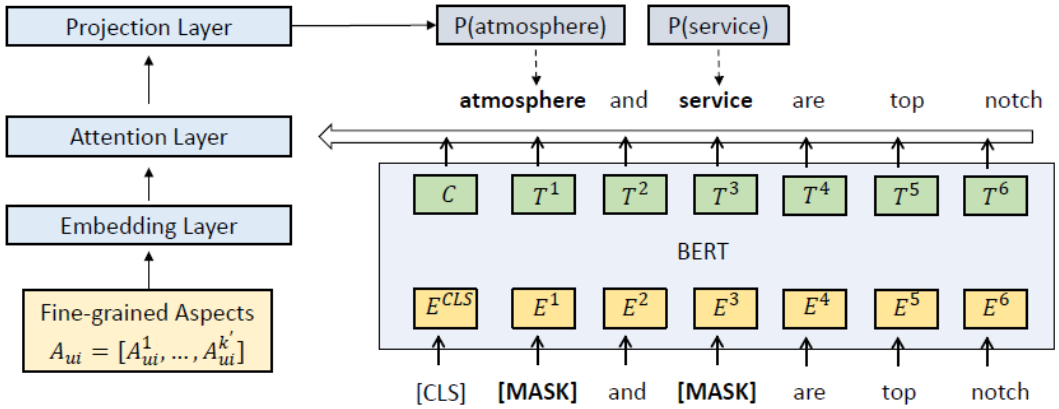

图 3-2 分词条件屏蔽语言模型的结构

遮蔽语言模型序列解码器。我们使用预训练 BERT 模型中的遮蔽语言模型作为序列解码器，并增加对方面编码器输出的注意力。如图 3-2 所示，解码器的输入是一个被遮蔽的理由$J_{u,i}^M = \{w_1, \dots, w_T\}$，其中多个词语被替换为[MASK]。解码器的输出$\mathbf{T} \in \mathbb{R}^{T \times n}$随后被送入注意力层，与编码器的输出计算注意力分数：

$$a_t^3 = \sum_{j=1}^{K'} \alpha_{tj}^3 \mathbf{A}_j$$

$$\alpha_{tj}^3 = \exp\left(\tanh\left(v_\alpha^{3\mathrm{T}}\left(W_\alpha^3\left[\mathbf{A}_j; \mathbf{T}_t\right] + b_\alpha^3\right)\right)\right)/Z$$

然后，注意力向量a_t^3与解码器在时间步t的隐藏状态连接，并送入线性投影层以获得输出词分布P。时间步t的词w的输出概率由下式给出：

$$p(w_t) = \tanh\left(W_2 \cdot [T_t;\ a_t^3] + b2\right)$$

其中w_t是时间步 t 的目标词。

遮蔽程序。原始 BERT 论文应用了一个固定比率（15%）来决定是否遮蔽一个词语。不同于他们的方法，我们采用更高的比率来遮蔽细粒度方面，因为它们在理由中更为重要。具体来说，如果我们遇到一个细粒度方面，我们将 30%的时间用[MASK]标记替换它；对于其他词语，则用 15%的时间用[MASK]标记替换。

在训练期间，模型只会预测这些被遮蔽的词语，并对它们计算交叉熵损失。

表 3.1 ACMLM 不同迭代的生成输出示例

第 0 次迭代	universe [MASK] is extremely friendly and persona ##ble
第 5 次迭代	the [MASK] is extremely friendly and persona ##ble
第 10 次迭代	the [MASK] is extremely friendly and persona ##ble
第 15 次迭代	the staff are extremely cool and persona ##ble
第 20 次迭代	the staff are extra kind , persona ##ble

通过从遮蔽模板采样生成。接下来，我们讨论如何从训练好的 ACMLM 生成理由。我们采用[27]Wang and Cho (2019)的采样策略来生成理由。我们不是从全是[MASK]标记的序列生成，而是从关于目标项目的历史理由生成的遮蔽模板开始。这些遮蔽模板包括了关于项目的先验知识，并且可以加速采样收敛的速度。

表3.1展示了生成过程的一个例子。我们将模板序列X0初始化为（宇宙，[MASK]；...；##ble）长度为 T。在每次迭代 i 中，从$\{1, \ldots, T\}$中均匀随机抽取一个位置t_i，并将当前序列X^i中位置 ti 的词语（即$x_{t_i}^i$）替换为[MASK]。之后，我们获得x_{t_i}的条件概率为

$$p\left(x_{t_i} \middle| X_{\setminus t_i}^i = \frac{1}{Z\left(X_{\setminus t_i}^i\right)} \exp\left(1\mathrm{h}\left(x_{t_i}\right)^{\mathrm{T}}\ f_\theta\left(X_{\setminus t_i}^i\right)\right)\right)$$

其中，$1\mathrm{h}\left(x_{t_i}\right)^T$是一个索引$x_{t_i}$设置为 1 的独热向量，$X_{\setminus t_i}^i$是在$X^i$中位置$t_i$的词语被替换为[MASK]后我们获得的序列，$f_\theta\left(X_{\setminus t_i}^i\right)$是将$X_{\setminus t_i}^i$输入 ACMLM 后的输出，如方程（8）所示，$Z$是归一化项。然后我们从方程（9）中抽样$\tilde{x}_{ti}$，并通过$X^{i+1} = (x_1^i, \ldots, \tilde{x}_{ti}, \ldots, x_T^i)$构造下一个序列。重复此过程$N$次后，最终输出被视为生成输出。[2]

[2] We set N proportional to the length T of the initial masked template to prevent the generation diverging too much from the original template.

4 实验

4.1 数据集

通过我们提出的流程（第 2 节），我们从现有的评论数据构建了两个个性化理由数据集——Yelp 和 Amazon Clothing[3][4]。我们进一步过滤掉那些少于五个理由的用户。对于每个用户，我们随机保留他们所有理由中的两个样本来构建开发集和测试集。表 3.2 显示了我们两个数据集的统计信息。

4.2 基线

对于自动评估，我们考虑了三个基线模型：Item-Rand 是一个随机从项目的历史理由中选择一个理由的基线模型。LexRank 是一个在文本摘要中广泛使用的强大的无监督基线模型[11]（Erkan and Radev, 2004）。给定一个项目的所有历史理由，LexRank 可以选择一个理由作为摘要。然后我们将其作为所有用户的理由。Attr2Seq[10]（Dong et al., 2017）是一个使用属性（即用户和项目身份）作为输入的 Seq2Seq 基线模型。

表 3.2 我们的数据集统计信息

数据集	训练集	开发集	测试集	用户数量	项目数量	方面数量
Yelp	1,219,962	115,907	115,907	115,907	51,948	2,041
亚马逊衣服	202,528	57,947	57,947	57,947	50,240	581

表 3.3 自动评估性能

数据集	Yelp				亚马逊衣服			
模型	BLEU-3	BLEU-4	Distinct-1	Distinct-2	BLEU-3	BLEU-4	Distinct-1	Distinct-2
Item-Rand	0.44	0.15	2.766	20.151	1.62	0.68	2.4	11.853
LexRank	2.29	0.92	1.738	8.509	3.48	2.25	2.407	14.956
Attr2seq	**7.89**	0	0.049	0.095	1.72	0.56	0.076	0.352
Ref2Seq	4.38	**2.45**	0.188	1.163	8.78	5.67	0.141	1.24
AP-Ref2Seq	3.39	1.83	0.326	2.094	**13.91**	**12.5**	0.557	3.661
Ref2Seq (Top-k)	1.63	0.7	0.818	11.927	3.96	2.13	0.697	**10.858**
ACMLM	0.7	0.28	**1.322**	**14.319**	2.42	1.59	**0.942**	9.312

默认情况下，所有模型在生成时使用束搜索。最近，有研究显示采样方法的生成输出在高熵任务上更加多样化和适用[14]（Holtzman et al., 2019）。为此，我们在实验中探索了另一种解码策略——“Top-k 采样”[25]（Radford et al., 2019），并包括了我们模型的一

[3] https://www.yelp.com/dataset/challenge

[4] http://jmcauley.ucsd.edu/data/amazon

个变体：Ref2Seq（Top-k）。[5]

对于人类评估，我们包括了两个基线：Ref2Seq（Review）和 Ref2Seq（Tip），两者都是与 Ref2Seq 模型相同但分别训练在原始评论和提示数据上的模型。与这两个基线的比较展示了在我们注释的数据集上训练倾向于生成更适合作为理由的文本。

4.3 实现细节

我们使用 PyTorch[6]来实现我们的模型。对于 Req2Seq 和 AP-Ref2Seq，我们将隐藏层大小和词嵌入大小设为 256。我们为编码器应用 0.5 的 dropout 率，为解码器应用 0.2 的 dropout 率。理由引用的大小 lr 设为 5，用户个性和项目简介中细粒度方面的数量 K 设为 30。我们使用 Adam 优化器进行训练，学习率为 2e-4，并在达到 20 个 epoch 或开发集上的困惑度不再改善时停止训练。对于 ACMLM，我们基于 HuggingFace 的 BERT 实现构建模型。[7]我们使用预训练的'Bert-base'模型初始化解码器，并将最大序列长度设为 30。我们使用学习率为 2e-5 的 Adam 优化器训练模型 5 个 epoch。对于使用束搜索的模型，我们将束大小设为 10。对于使用“top-k”采样的模型，我们将 k 设为 5。对于 ACMLM，我们使用等于初始序列长度的烧录步骤。我们的数据和代码可在线获取。[8]

表 3.4 人类评价绩效，其中 R、I、D 分别代表相关性（Relevance）、信息性（Informativeness）和多样性（Diversity）。多样性。

模型	相关性 (R)	信息量 (I)	多样性 (D)
Ref2Seq (Review)	3.02	2.39	2.10
Ref2Seq (Tip)	3.25	2.35	2.34
Ref2Seq	3.87	3.13	2.96
Ref2Seq (Top-k)	3.95	3.34	3.39
ACMLM	3.23	3.29	3.42

4.4 自动评估

对于自动评估，我们使用 BLEU、Distinct-1 和 Distinct-2[15]（Li et al., 2015）来衡量我们模型的性能。如表 3.5 所示，我们的基于参考的模型在两个数据集上获得了最高的 BLEU 分数，除了 Yelp 上的 BLEU-3。这证实了 Ref2Seq 能够捕捉用户和项目内容，与未个性化的模型如 LexRank 以及不利用历史理由的个性化模型如 Attr2Seq 相比，生成最相关的内容。

另一方面，最近的研究报告称，在开放域生成任务中，获得更高多样性分数的模型

[5] 在每个时间步长内，下一个词将从前 k 个可能的下一个词块中抽取。k 个可能的下一个词块中抽取。

[6] http://pytorch.org/docs/master/index.html

[7] https://github.com/huggingface/pytorch-pretrained-BERT

[8] https://github.com/nijianmo/recsys justification.git

将在基于重叠的度量（例如 BLEU）上得分较低[3][12]（Baheti et al., 2018; Gao et al., 2018）。我们对我们的个性化理由生成任务做出了类似的观察。如表 3.3 所示，基于采样的方法 Ref2Seq（Top-k）和 ACMLM 在 Distinct-1 和 Distinct-2 上得分更高，而它们的 BLEU 分数低于基于束搜索的 Seq2Seq 模型。因此，我们还进行了人类评估，以验证我们提出方法的生成质量。

表 3.5 对 Yelp 数据集上三家企业的不同模型生成的理由进行比较。

模型	Shake Shack	Teharu Sushi	MGM Grand Hotel
真实情况	汉堡很好	卷饼很棒，典型的卷饼并不是很多特色	房间非常干净舒适
LexRank	很棒的汉堡和薯条	寿司？	很棒的房间
Ref2Seq (评论)	我爱 Trader Joe's，我爱 Trader Joe's	食物很好，服务很棒	我爱这个地方！食物总是很好，服务总是很棒
Ref2Seq (提示)	这个地方很棒，爱这个地方，来这里	爱这个地方	来这里
Ref2Seq	这个地方有一些最好的汉堡	寿司很美味	房间很好
Ref2Seq (Top-k)	薯条很惊艳	新鲜美味的寿司	开放酒店几小时
ACMLM	早餐三明治总体上非常满足	整体上有趣的体验，半价寿司	家庭式晚餐，长时间的购物之旅到拉斯维加斯，家庭用餐，便宜的午餐

4.5 手动评估

我们对三个方面进行了手动评估：（1）相关性衡量生成的输出是否包含与项目相关的信息；（2）信息量衡量生成的理由是否包含对用户有帮助的具体信息；以及（3）多样性衡量与其他理由相比，生成的输出有多么独特。

我们专注于 Yelp 数据集，并从表 3.4 中展示的五个模型中抽取 100 个生成的示例。要求手动评注者为每个指标给出一个[1,5]（最低到最高）范围内的分数。每个示例至少由三名评注者评分。结果显示，与其他模型相比，Ref2Seq（Top-k）和 ACMLM 在多样性和信息量上获得了更高的分数。

4.6 定性分析

这里我们研究以下两个定性问题：

RQ1：训练数据和方法如何影响生成？如表 3.5 所示，训练在评论和小费数据上的

模型倾向于生成通用短语（例如“我爱这个地方”），这些短语通常不包括帮助用户做决策的信息。训练在理由数据集上的其他模型倾向于提及具体信息（例如不同方面）。LexRank 倾向于生成相关但内容较短的内容。同时，基于采样的模型能够生成更多样化的内容。

RQ2：方面规划如何影响生成？为了缓解多样性和相关性之间的权衡，一种方法是在生成过程中增加更多约束，例如受限束搜索[1]（Anderson et al., 2017）。在我们的工作中，我们通过整合方面规划来指导生成，从而扩展了我们的基础模型 Ref2Seq。如表 3.6 所示，大多数计划中的方面都出现在 AP-Req2Seq 生成的输出中。

如同表 3.6，展示从 AP-Ref2Seq 生成的理由。这些计划中的方面是从用户个人特征中随机选择的。

表 3.6 从用户个人特征中随机选择的方面对应的 AP-Ref2Seq 生成的理由

数据集	方面	生成的理由
Yelp	用餐	用餐环境很好
Yelp	糕点	糕点相当不错
Yelp	鸡肉	鸡肉炒饭是最好的
Yelp	三明治	手撕猪肉三明治是菜单上最好的
Amazon-Clothing	产品	产品很棒，快速发货
Amazon-Clothing	价格	设计漂亮，价格合理
Amazon-Clothing	皮革	舒适的皮革运动鞋
Amazon-Clothing	步行	结实，适合城市步行的鞋子

5 相关工作

解释性推荐 近年来，如何提高推荐系统的解释性成为一个研究热点。Catherine 和 Cohen（2017）[6]学习评论文本的潜在表示来预测评分。这些表示随后被用来为特定的用户和项目对找到最有帮助的评论。另一个流行的方向是生成文本来证明推荐。Dong et al（2017）[10]提出了一个属性到序列模型来生成利用分类属性的产品评论。Ni et al（2017）[23]开发了一种多任务学习方法，考虑了协同过滤和评论生成。Li et al（2019b）[18]通过考虑“个人特征”信息生成提示，这可以捕捉用户的语言风格和项目的特点。然而，这些工作使用整个评论或提示作为训练实例，可能不适当，因为评论文本的质量问题。最近，Liu et al（2019）[20]提出了一个框架来生成文本分类的细粒度解释。为了实现人类可读的解释标签，他们从一个提供评分和用户编写的细粒度摘要的网站构建了一个数据集。不幸的是，大多数网站不提供这种细粒度信息。另一方面，我们的工作从评论中识别理由，将它们作为训练示例，并通过广泛的实验表明这些是可解释推荐的更好数据源。

多样性意识的自然语言生成 多样性是自然语言生成系统的一个重要方面。近期的工作集中在消化先验知识以提高生成多样性。Yao et al（2019）[33]提出了一种在故事生

成中结合计划故事情节的方法。Li et al（2019a）[17]开发了一种考虑方面的粗到细评论生成方法。他们为评论中的每个句子预测一个方面，以捕捉内容流。给定方面后，生成一系列句子草稿，解码器将填充每个草稿的槽位。在对话系统中，一些工作研究了从历史回应中提取模板的框架，然后编辑这些模板形成新的回应（Weston et al，2018；Wu et al，2018）[29][32]。同样，提取和编辑范式也在 NLG 的风格转换任务中进行了研究（Li et al，2018）[16]。Wu et al（2019）[30]提出了一个用于非平行情感转换的属性感知遮蔽语言模型。他们首先遮蔽情感词汇，然后训练一个遮蔽语言模型为目标情感填充遮蔽位置。在这项工作中，我们也引入了条件遮蔽语言模型，但考虑了更细粒度的方面。

6 结论

在这项工作中，我们研究了个性化理由生成的问题。为了构建高质量的理由数据集，我们提供了一个注释数据集，并提出了一个从大量评论语料中提取理由的流程。为了生成令人信服且多样化的理由，我们开发了两个模型：（1）Ref2Seq 利用历史理由作为生成过程中的参考；（2）ACMLM 是一个基于预训练遮蔽语言模型的面向方面的条件模型。我们的实验表明，与基线相比，Ref2Seq 在 BLEU 分数上得分更高，ACMLM 在多样性分数上得分更高。人类评估表明，基于参考的模型获得了高相关性分数，基于采样的方法产生了更多样化和信息丰富的输出。最后，我们展示了方面规划是引导生成以产生个性化和相关理由的有前景的方式。

参考文献

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2017. Guided open vocabulary image captioning with constrained beam search. In EMNLP.

[2] Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In EMNLP.

[3] Ashutosh Baheti, Alan Ritter, Jiwei Li, and William B. Dolan. 2018. Generating more interesting responses in neural conversation models with distributional constraints. In EMNLP.

[4] Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. 2015. Better document-level sentiment analysis from RST discourse parsing. In EMNLP.

[5] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, 5:135–146.

[6] Rose Catherine and William W. Cohen. 2017. Transnets: Learning to transform for recommendation. In RecSys.

[7] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In EMNLP.

[8] JacobWillem Cohen. 1960. A coefficient of agreement for nominal scales.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In NAACL.

[10]Li Dong, Shaohan Huang, Furu Wei, Mirella Lapata, Ming Zhou, and Ke Xu. 2017. Learning to generate product reviews from attributes. In EACL.

[11]Güneş Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res., 22:457–479.

[12]Jun Gao, Wei Bi, Xiaojiang Liu, Junhui Li, and Shuming Shi. 2018. Generating multiple diverse responses for short-text conversation. In AAAI.

[13]Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. Neural Computation, 9:1735–1780.

[14]Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. CoRR, abs/1904.09751.

[15]Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B. Dolan. 2015. A diversity-promoting objective function for neural conversation models. In HLT-NAACL.

[16]Juncen Li, Robin Jia, He He, and Percy S. Liang. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer. In NAACL-HLT.

[17]Junyi Li, Wayne Xin Zhao, Ji-Rong Wen, and Yang Song. 2019a. Generating long and informative reviews with aspect-aware coarse-to-fine decoding. In ACL.

[18]Piji Li, Zihao Wang, Lidong Bing, and Wai Lam. 2019b. Persona-aware tips generation. In WWW.

[19]Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. 2017. Neural rating regression with abstractive tips generation for recommendation. In SIGIR.

[20]Hui Liu, Qingyu Yin, and William Yang Wang. 2019. Towards explainable NLP: A generative explanation framework for text classification. In ACL.

[21]William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: toward a functional theory of text.

[22]Elman Mansimov, Alex Wang, and Kyunghyun Cho. 2019. A generalized framework of sequence generation with application to undirected sequence models. ArXiv, abs/1905.12790.

[23]Jianmo Ni, Zachary C. Lipton, Sharad Vikram, and Julian J. McAuley. 2017. Estimating reactions and recommending products with generative models of reviews. In IJCNLP.

[24]Jianmo Ni and Julian McAuley. 2018. Personalized review generation by expanding phrases and attending on aspect-aware representations. In ACL.

[25]Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

[26]Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In NIPS.

[27]Alex Wang and Kyunghyun Cho. 2019. Bert has a mouth, and it must speak: Bert as a

markov random field language model. CoRR, abs/1902.04094.

[28]Yizhong Wang, Sujian Li, and Jingfeng Yang. 2018. Toward fast and accurate neural discourse segmentation. In EMNLP.

[29]Jason Weston, Emily Dinan, and Alexander H. Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. In SCAI@EMNLP.

[30]Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Mask and infill: Applying masked language model to sentiment transfer. In IJCAI.

[31]Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. CoRR, abs/1609.08144.

[32]Yu Wu, Furu Wei, Shaohan Huang, Yunli Wang, Zhoujun Li, and Ming Zhou. 2018. Response generation by context-aware prototype editing. In AAAI.

[33]Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In AAAI.

[34]Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In SIGIR.